OCR
Oxford Cambridge and RSA

An OCR endorsed teaching and learning tool

OCR A Level Computer Science

H446 – Paper 1

1

Primitive data types, binary and hexadecimal

Unit 6
Data types

PG ONLINE

Objectives

- List and define primitive data types
- Represent positive integers in binary and hexadecimal
- Convert between binary, hexadecimal and denary

Primitive data types

- A primitive data type is one which is provided by a programming language
- They include:
  - integer a whole number such as 34, 0, -1, 567432
  - real/float a number with a fractional part such as:
    3.142, 7.0, -67.5
- Usually, a composite data type such as an array is not considered to be a primitive data type
  - Can you think of some more primitive data types?

Primitive data types

- A primitive data type is one which is provided by a programming language
- They include:
  - integer — a whole number such as 34, 0, -1, 567432
  - real/float — a number with a fractional part such as 3.142, -67.5
  - Boolean — can only take the value True or False
  - character — a letter, number or special symbol such as “a”, “A”, “6”, “&”, %”
  - string — anything enclosed in quote marks, for example “Jason”, “01798 158794”, “This is a string”

How are different data types held?

- All data types are held in binary
- Without knowing what the data type is, it is not possible to say what a particular bit pattern represents

```
00101000 111110001 01010111 00100100
```

- This could be one integer, 4 integers, a real number, a string, 4 characters,...
- or even a sound, a pixel or a tiny piece of a graphic

Binary number system

- The binary number system uses only two digits,
  0 and 1
- To understand how this works, it is helpful to look first at the fundamentals of the **denary** or **decimal** system, which uses digits 0..9

Number symbols

- The **denary** number system uses a combination of just ten symbols to represent any number
- Number systems are referred to by their **base**; that is, the number of symbols used to construct values
- Denary is **base 10** and the base may be referred to as a subscript
  - In denary the number 11, for example, may be written as 11_{10}
  - In binary, it would be written as 11_2 and in hexadecimal as 11_{16}

Base 2

- Numbers which use **base 2** are commonly referred to as **binary** numbers
- Can you see a pattern in the binary values?

Denary	Binary
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011

Place value

- In any counting system, the **position** of the value determines its contribution to the overall total

Base^{position}	10^2	10^1	10^0
Place value	100	10	1
Number	1	8	3
Digit value	100x1	10x8	1x3

- Therefore 183_{10} represents the value of one hundred, eight tens and three units

The binary system

- Binary works the same way, except the base used is 2 instead of 10

Base^{position}	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Place value	128	64	32	16	8	4	2	1
Number	1	0	1	1	0	1	1	1
Digit value	128x 1	0	32x 1	16x 1	0	4x1	2x1	1x 1

- Therefore $1\ 0\ 1\ 1\ 0\ 1\ 1\ 1_2$ represents $128 + 32 + 16 + 4 + 2 + 1$ which is equivalent to 183_{10}

Converting from denary

- The place value table is useful to aid conversion from denary to binary
  - Convert 159_{10} to binary

Place value	128	64	32	16	8	4	2	1
Number	1	0	0	1	1	1	1	1

1. Can 128 be used to make some of 159?

2. If it can, use it by setting bit value to 1, (0 if not)

3. If value used subtract from starting value, eg. 159 – 128 = 31

4. Repeat using value from step 3 until nothing left

Activity

- Complete **Task 1** in **Worksheet 1**

Hexadecimal – Base 16

- Number systems with bases above 10 often still use 0 to 9 but require extra symbols for values past 9
- An example of this is **base 16** which is referred to as the **hexadecimal** number system
- Hexadecimal uses letters for the values 10 to 15

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

0123456789ABCDEF

Hexadecimal values

- The rules of place value work in the same way:

Base^{position}	16^2	16^1	16^0
Place value	256	16	1
Number	3	F (or 15)	5
Digit value	256 x 3	16 x 15	1 x 5

- Therefore the equivalent of $3F5_{16}$ in denary is (256x3) + (16x15) + 5 = 768 + 240 + 5 = 1013_{10}
- What is $A3_{16}$ in denary?

Denary to hex conversion

- Converting to two-digit hexadecimal numbers
- Divide the number by 16, and add the remainder

43	÷	16	=	2 remainder 11
1 1			=	B
			=	2B

- What is 27_{10} in hexadecimal?

Significant powers

- Work out the following powers of 2:

$2^0 =$

$2^1 =$

$2^2 =$

$2^3 =$

$2^4 =$

- Why is the answer to 2^4 significant?

Significant powers

- Work out the following powers of 2:

$2^0 = 1$

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 =$ 16

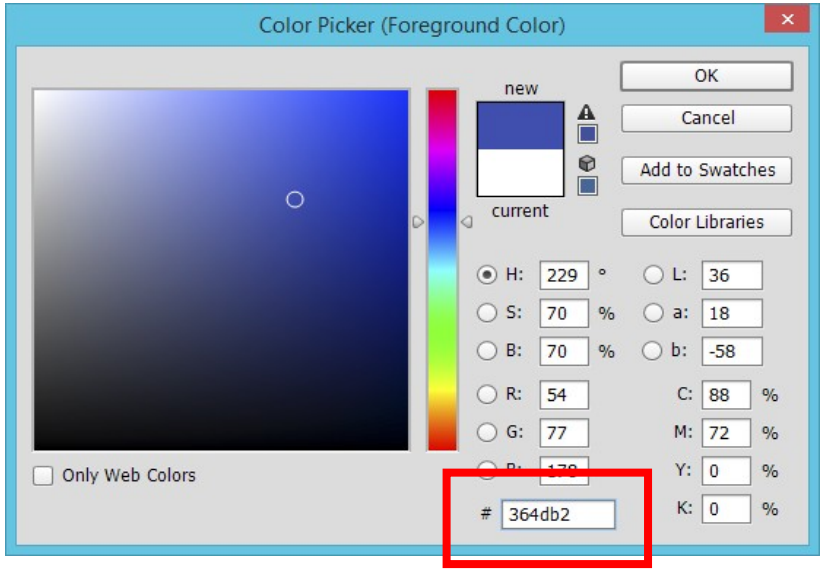
- Why is the answer to 2^4 significant?

Significance of 16

- 16 is the fourth power of 2
- This means that base 16 numbers can be translated from 4 consecutive bits of a binary value
- This makes it simple to translate binary numbers into hexadecimal values and back again

	8 4 2 1	8 4 2 1
216 =	1 1 0 1	1 0 0 0
=	D	8

Hex colour codes



- The hex code for this colour is 36 4D B2
  - What does this mean?
  - How can you increase the amount of red, using the hex code?

Activity

- Complete **Task 2** on **Worksheet 1**

Why use hex?

- A hexadecimal value is much easier to read and remember than a string of binary digits
- It is quicker to write or type, since a hex digit takes up only one character, not four
- There is less chance of making an error when typing hex characters than a string of 1s and 0s
- It is used to define colours, in MAC addresses, in assembly languages and machine code
- It is very easy to convert to and from binary

Plenary

- Denary numbers are easy to understand but they are not easy to use in electronics
- Binary is simple to use with circuitry but is difficult to read by humans
- Hexadecimal makes binary easier to read and easier to transcribe (copy) accurately

Copyright

Licence agreement